

REMARKS

The Office Action mailed February 24, 2009 has been received and reviewed. A Request For Continued Examination has been filed concurrently with this Amendment. In the Office Action, each of claims 1-9, 11-14, 16-26 and 28-33 stands rejected. Claims 1, 9, 17, and 26 have been amended and claims 5, 18, 19, 23, 27, 31, and 32 have been canceled herein. As such, claims 1-4, 6-9, 11-14, 16-17, 20-22, 24-26, 28-30 and 33 remain pending, with claims 1, 9, 17 and 26 being in independent form. Care has been exercised to introduce no new subject matter. Reconsideration of the above-identified application in view of the above amendments and the following remarks is respectfully requested.

Objections

In the Office Action, claim 9 was objected to for improperly reciting the phrase "computer readable storage medium. . ." In response, Applicants have corrected the wording of claim 9, as suggested in the Office Action. Withdrawal of this objection is requested.

Rejections based on 35 U.S.C. § 103

Also in the Office Action, claims 1-9, 11-14, 16-26 and 28-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Walker (U.S. Publication No. 2004/0015840, hereinafter the "Walker reference") in view of Bhansali (U.S. Publication No. 2002/0169999, hereinafter the "Bhansali reference"). Because claims 5, 18, 19, 23, 27, 31, and 32 have been canceled herein, Applicants submit that the rejections of these claims has been rendered moot. As for the claims that remain pending, Applicants submit that combination proposed in the Office Action does not support a prima facie case of obviousness. As such, Applicants respectfully traverse this rejection, as hereinafter set forth.

A) Applicable Authority

To sustain a rejection of a claim under 35 U.S.C. § 103(a), the Examiner must find that a preponderance of the evidence supports a finding of obviousness. The Examiner bears the initial burden of showing that the reference teachings establish a *prima facie* case of obviousness. “In view of all factual information, the examiner must . . . make a determination whether the claimed invention ‘as a whole’ would have been obvious at that time to that person.” MPEP § 2142 (explaining that “[t]he key to supporting any rejection under 35 U.S.C. 103 is the clear articulation of the reason(s) why the claimed invention would have been obvious”). In making that determination, the Examiner must consider every word in each claim. *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

Recently, the Supreme Court indicated that “it will be necessary for [the Office] to look at interrelated teachings of multiple [prior art references]; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by [one of] ordinary skill in the art . . . to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the [patent application].” *KSR v. Teleflex*, 127 S. Ct. 1727 (2007). However, if the references do not “expressly or impliedly suggest the claimed invention,” the examiner must present “a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references.” MPEP § 706.02(j) and § 2142, quoting *Ex parte Clapp*, 227 USPQ 972, 972 (Bd. Pat. App. & Inter. 1985).

B) Walker in view of Bhansali fail to disclose each and every limitation of claim 1, as amended herein

Independent claim 1, as amended herein, is generally directed to a compiling system embodied on a computer readable storage medium for compiling a markup language file

into an executable application. The compiling system includes a parser for parsing the markup language file and providing the compiling system with detailed token information. The markup language file is associated with at least one C# file. The compiling system also includes a code generator for generating a language-independent tree of code expressions based on the token information. The code generator receives the code in the at least one C# file inside the markup language file. The code expressions represent the markup file as a class. The compiling system further includes a compiler for compiling the code expressions to create the executable application. The compiler determines an appropriate code provider for generating code in an appropriate language in response to the language-independent tree of code expressions.

The Walker reference, in contrast, discloses a method and mechanism for converting between JAVA classes and XML. *See Walker, Abstract.* In particular, the method disclosed in the Walker reference allows programmers to “convert easily between JAVA and XML representations of data while working exclusively with classes from the application domain. The invention enables developers to easily add XML support for complex hierarchies of any user-defined class, JAVA primitives (int, float, boolean, etc.) and wrapper classes (Integer, Boolean, Float, etc.) as well as collections and arrays of such objects.” *Id.* at ¶ [0018].

The Walker reference fails to disclose, however, many features of independent claim 1, as amended herein. For example only, the Walker reference fails to disclose “a code generator for generating a language-independent tree of code expressions based on the token information.” *See Office Action, p. 5.* Because the Walker reference fails to disclose the more general feature of a “code generator for generating a language independent tree of code expressions,” the Walker reference must also fail to disclose the claimed “a compiler for compiling the code expressions to create the executable application, wherein the compiler

determines an appropriate code provider for generating code in an appropriate language in response to the language-independent tree of code expressions.” Similarly, the Walker reference fails to disclose a markup language file being “associated with at least one C# file.” For at least these reasons, the Walker reference fails to disclose each and every limitation of independent claim 1, as amended herein.

The Bhansali reference discloses a compiler that completes compilation upon identifying an unresolvable input during compilation instead of aborting compilation. Bhansali, Abstract. The invention of Bhansali achieves this by incorporating an exception-throwing instruction(s) in the output code of the compiler. *Id.* at ¶ [0020]. The Bhansali reference fails to cure the many deficiencies stated with regard to the Walker reference.

For at least the reasons identified above, Applicants submit that claim 1, as amended herein, is not obvious over the combination of Walker in view of Bhansali. For this reason, Applicants submit that claim 1 is allowable over the cited references of record. Because claims 2-4 and 6-8 depend, either directly or indirectly, from allowable claim 1, these claims also stand in a condition for allowance and such favorable action is hereby requested.

C) Walker in view of Bhansali fail to disclose each and every limitation of claims 9 and 17, as each is amended herein

Independent claim 9, as amended herein, is generally directed to a compiling system embodied on a computer readable storage medium for compiling a markup language file into an executable application. The compiling system includes a parser for parsing the markup language file and providing the compiling system with detailed token information including non-code token information to the compiling system. The markup language file is associated with at least one C# file. The compiling system also includes a binary file generator for generating a

binary file from non-code token information. The binary file contains one record for each non-code token. The compiling system further includes a code generator for generating a language-independent code expression that represents the markup language file as a class. The compiling system still further includes an application generator for compiling the code files into an executable application. The application generator determines an appropriate code provider for generating code in an appropriate language in response to the language-independent tree of code expressions.

Independent claim 17, as amended herein, is generally directed to a method for compiling a markup language file into an executable application. The method includes receiving a markup language file. The method also includes receiving a C# file. The C# file being associated with the received markup language file. The method further includes parsing the markup language file into tokens and providing a compiling system with detailed information about the parsed tokens. The detailed information including inline code from the associated C# file. The method still further includes receiving a command to create an intermediate language application. The method also includes, upon receipt of the command to create an intermediate language application, generating a language-independent tree of code expressions based on the token information, wherein the code expressions represent the markup language file as a class. The method includes compiling the code expressions generated from the markup language file to create the executable application.

Applicants submit that claims 9 and 17 are allowable for some of those reasons identified with reference to independent claim 1. Because claims 11-14 and 16 depend, either directly or indirectly, from allowable claim 9, and because claims 20-22 and 24-25 depend, either directly or indirectly, from allowable claim 17, each of these claims also stand in a

condition for allowance and such favorable action is hereby requested.

D) Walker in view of Bhansali fail to disclose each and every limitation of claim 26, as amended herein

Independent claim 26, as amended herein, is generally directed to a method for compiling a markup language file into an executable application. The method includes receiving the markup language file. The method also includes receiving at least one code-behind file. The at least one code-behind file contains a user code snippet and is associated with the received markup language file. The method further parses the markup language file into tokens and providing a compiling system with detailed token information. The detailed token information includes non-code token information. The method still further includes receiving a command to create an application or library containing a binary tokenized representation of the markup language file. The method includes, in response to receiving the command to create an application or library containing a binary tokenized representation of the markup language file, generating a binary file from the non-code token information. The binary file contains one record for each non-code token. The method also includes generating a language-independent code expression that represents the markup language file as a class. The generated language-independent code expressions includes the user code snippet from the at least one code-behind file. The method further includes compiling the code expressions into an executable application. The compiling includes both the markup language file and the at least one code-behind file.

The Walker and Bhansali combination fails to disclose each and every limitation of claim 26, as amended herein. As such, Applicants submit that claim 26, as amended herein, is not obvious over the combination of Walker in view of Bhansali. For this reason, Applicants submit that claim 26 is allowable over the cited references of record. Because claims 28-30 and

32-33 depend, either directly or indirectly, from allowable claim 26, these claims also stand in a condition for allowance and such favorable action is hereby requested.

CONCLUSION

For at least the reasons stated above, claims 1-4, 6-9, 11-14, 16-17, 20-22, 24-26, 28-30 and 33 are now in condition for allowance. Applicants respectfully request withdrawal of the pending rejections and allowance of the claims. If any issues remain that would prevent issuance of this application, the Examiner is urged to contact the undersigned – 816-474-6550 or ahankel@shb.com (such communication via email is herein expressly granted) – to resolve the same. It is believed that no fee is due, however, the Commissioner is hereby authorized to charge any amount required to Deposit Account No. 19-2112.

Respectfully submitted,

/s/ Aaron E. Hankel /

Aaron E. Hankel
Reg. No. 60,663

AEHX/tq
SHOOK, HARDY & BACON L.L.P.
2555 Grand Blvd.
Kansas City, MO 64108-2613
816-474-6550